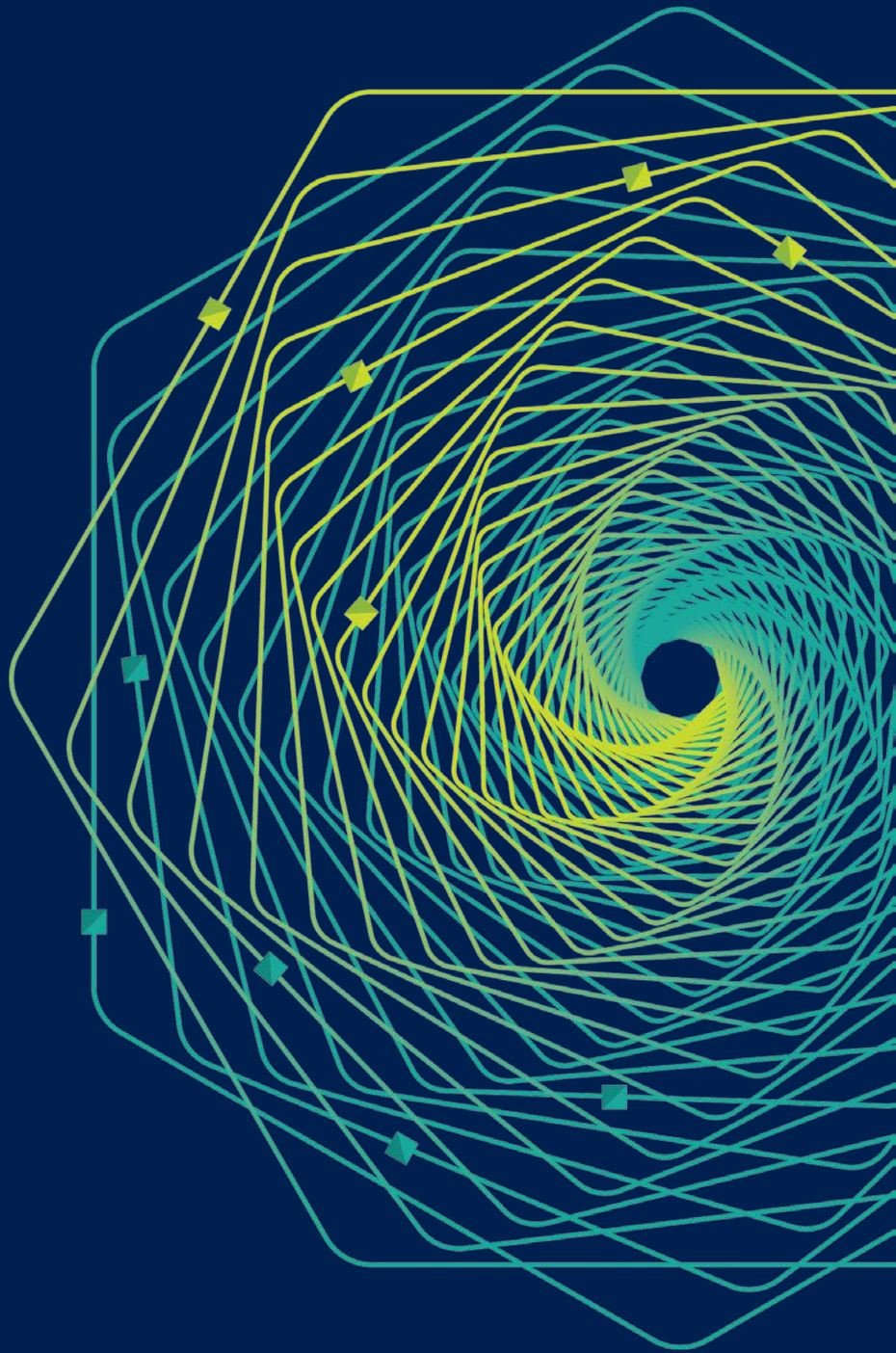
Microsoft

# Research Faculty Summit 2018

Systems | Fueling future disruptions

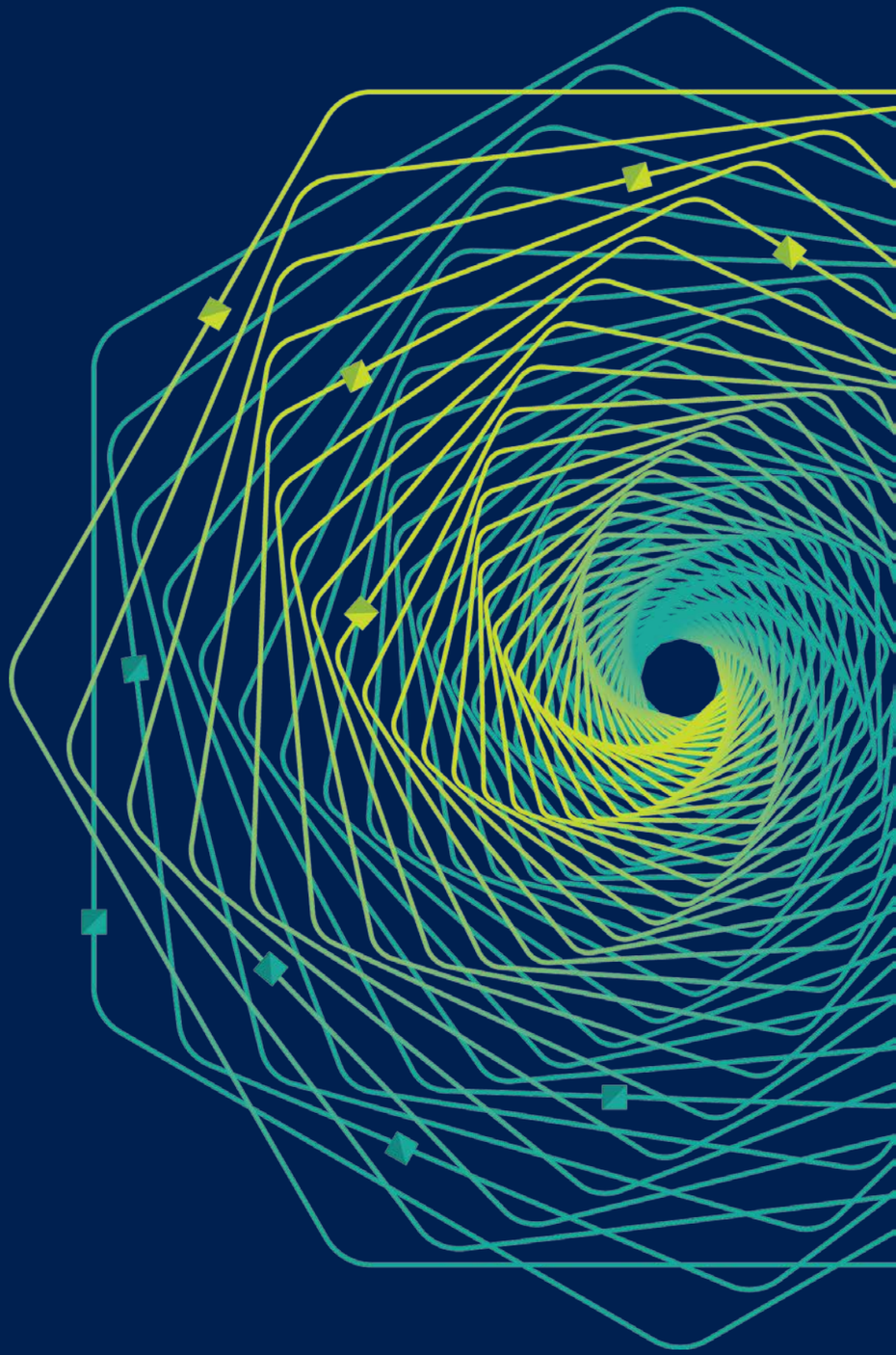


Microsoft

# Secure Speculative Execution Processors

Ilia Lebedev, Srini Devadas

With contributions from Victor Costan,
Vladimir Kiriansky, Saman Amarasinghe and Joel Emer



# Outline

- Violating isolation by exploiting speculative execution
- Defenses against cache timing attacks
- Secure enclaves in Intel SGX and MIT Sanctum

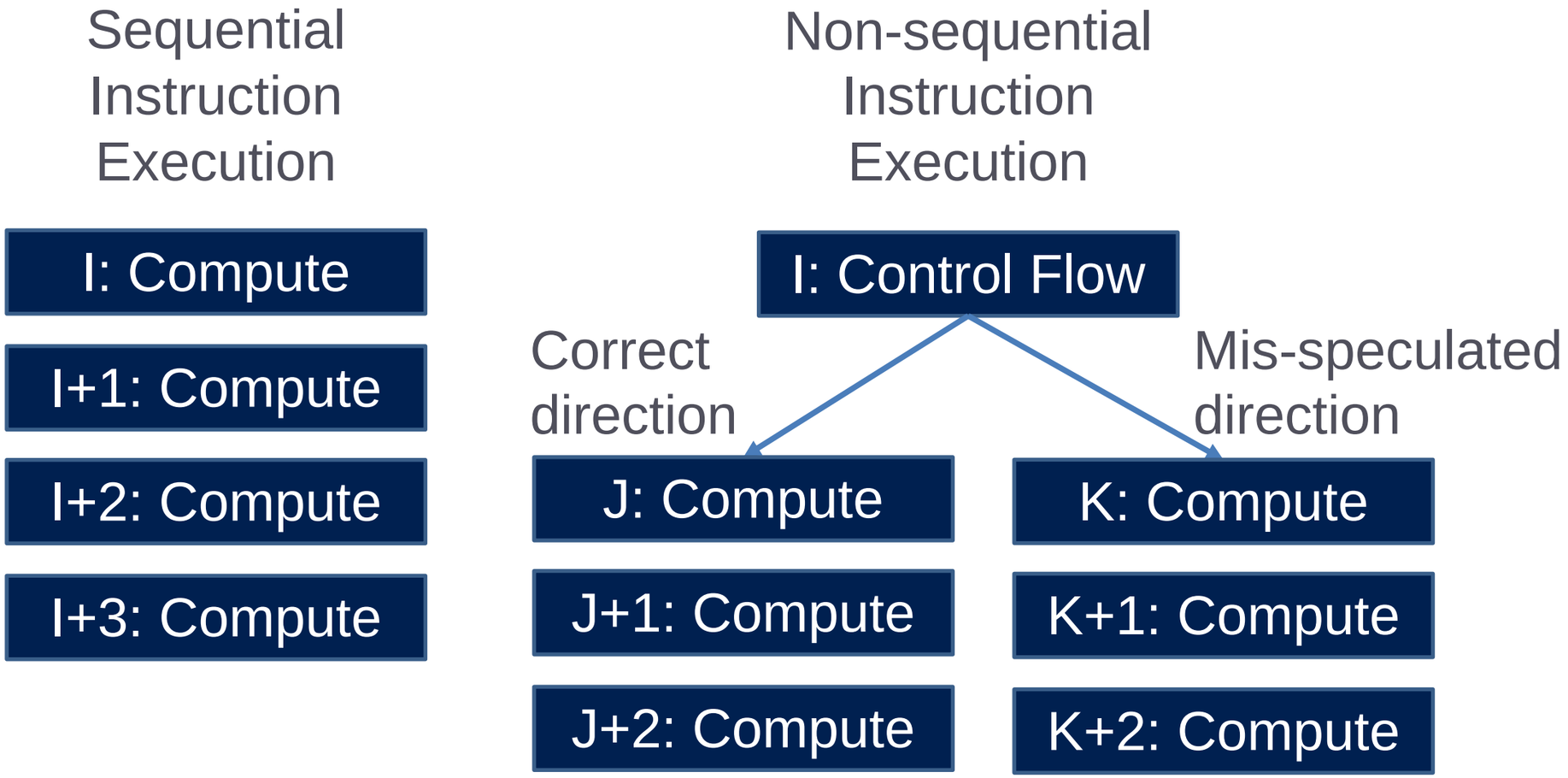
# Outline

- **Violating isolation by exploiting speculative execution**
- Defenses against cache timing attacks
- Secure enclaves in Intel SGX and MIT Sanctum

# Architectural Isolation

Fundamental to maintaining correctness and privacy!

# Control Flow Speculation for Performance



# Control Flow Speculation is insecure

Speculative execution does not affect architectural state → “correct”

… but can be observed via some “side channels” (primarily cache tag state)

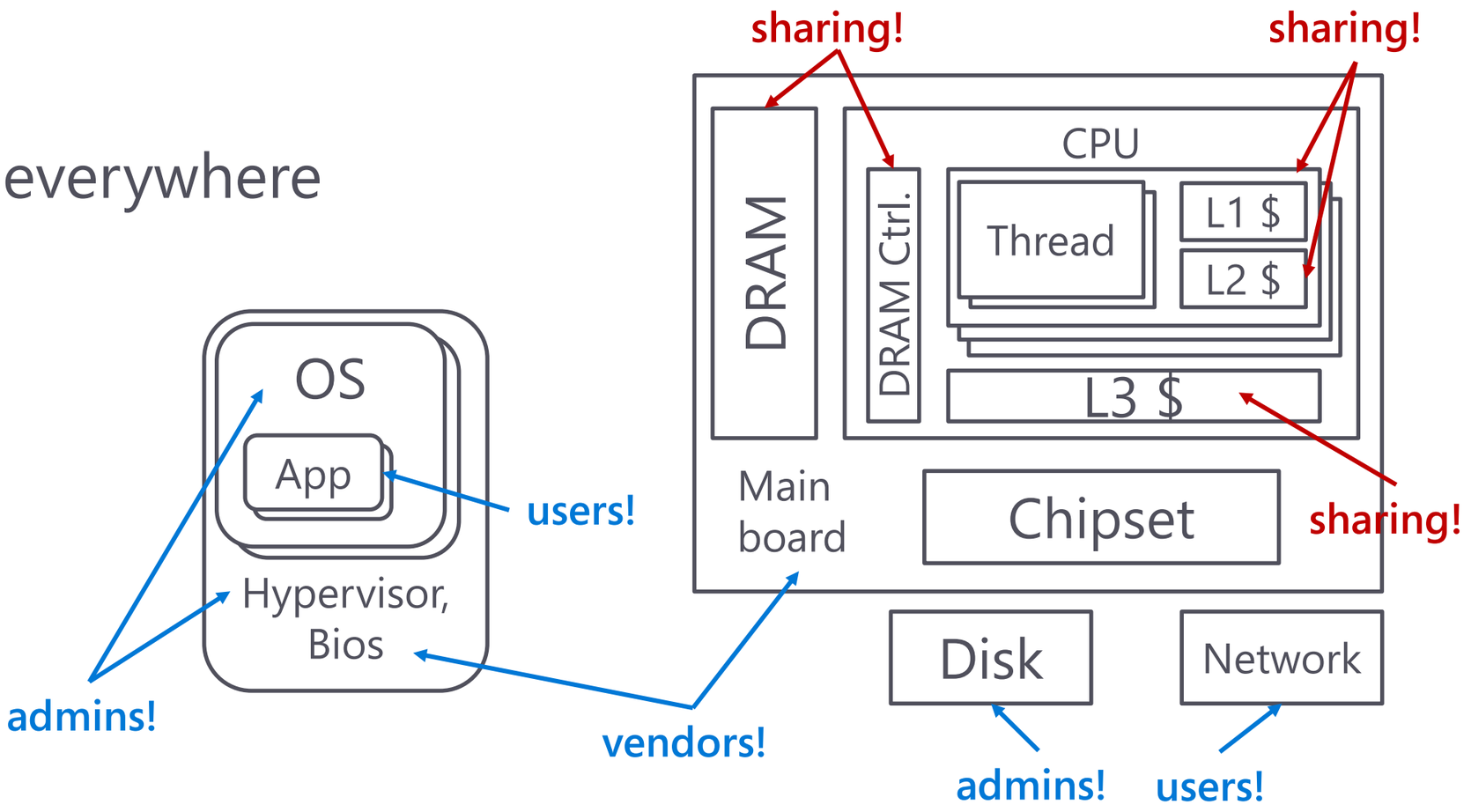
… and attacker can influence (mis)speculation →

branch predictor inputs not authenticated

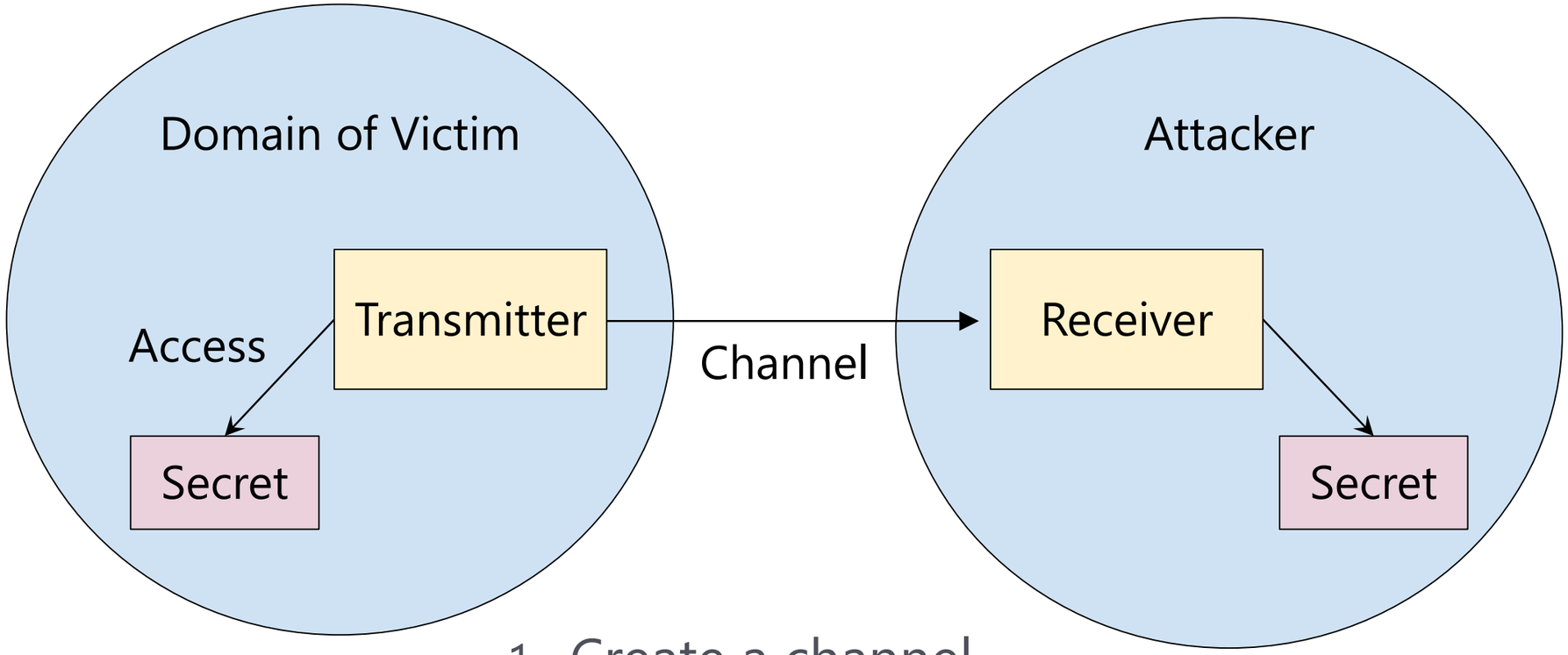
A huge, complex attack surface!

# Side Channels in the Wild

- Real systems: large, complex, cyberphysical (not secure)
- Spies potentially everywhere

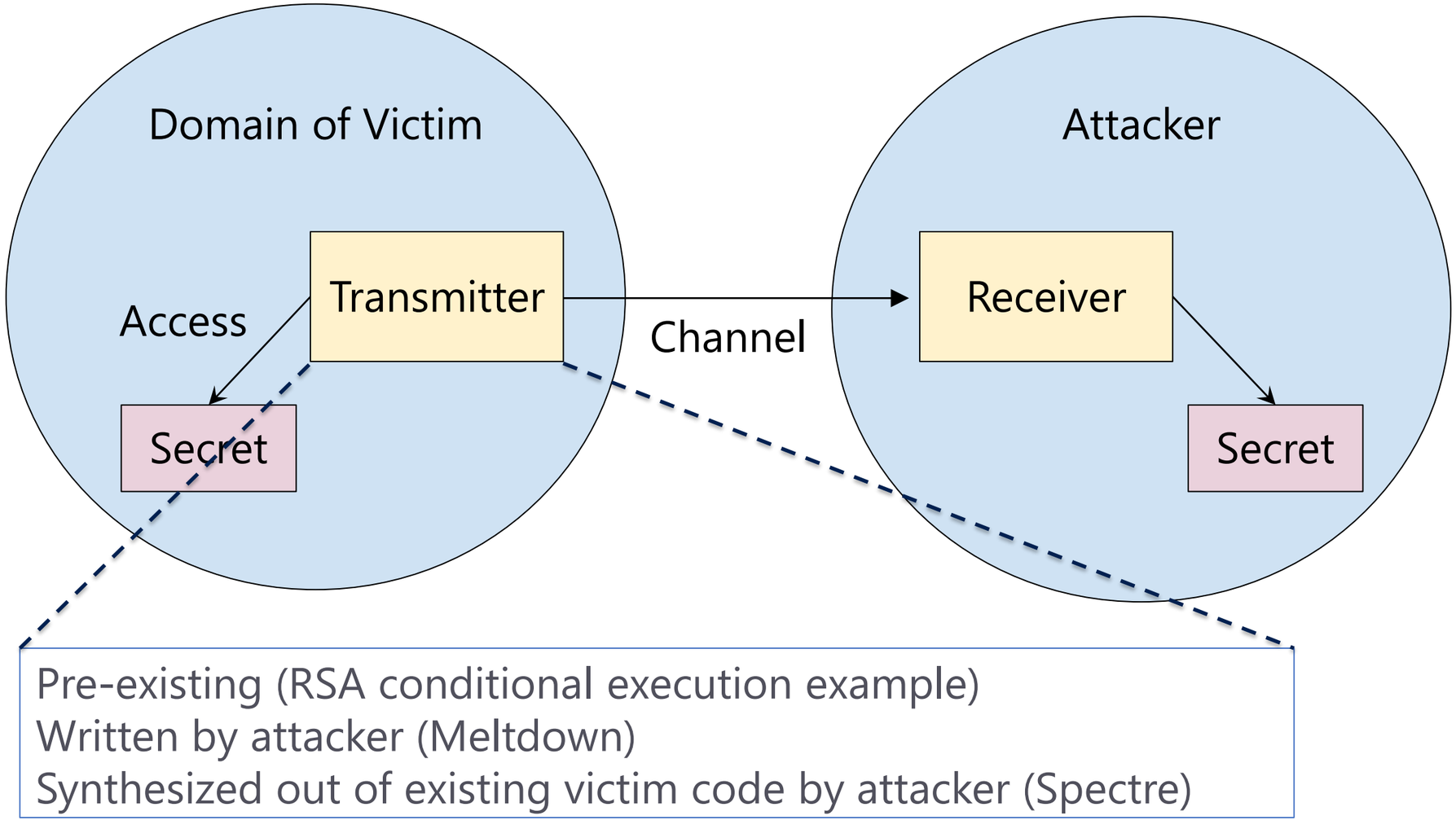


# Attack Schema




1. Create a channel
2. Create the transmitter
3. Launch the transmitter
4. Access the secret

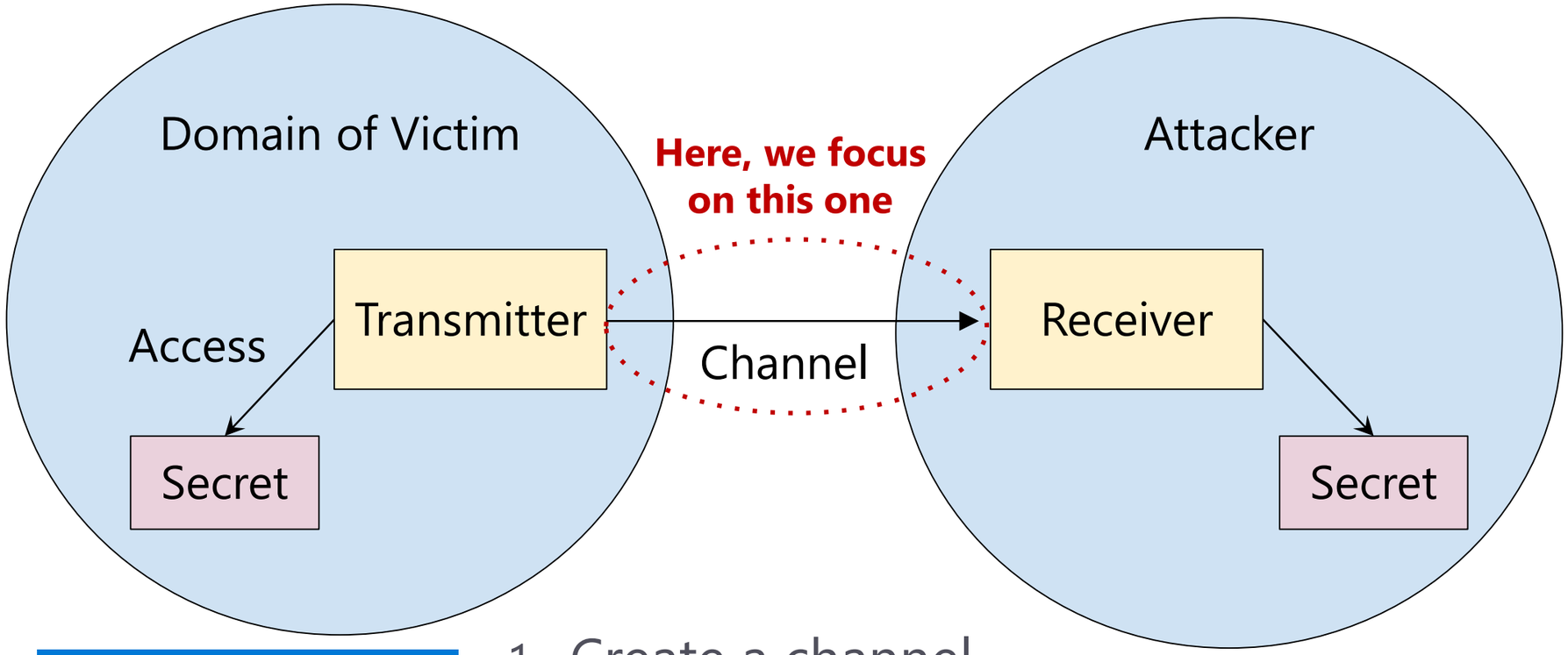
# Building a Transmitter



# Outline

- Violating isolation by exploiting speculative execution
- **Defenses against cache timing attacks**
- Secure enclaves in Intel SGX and MIT Sanctum

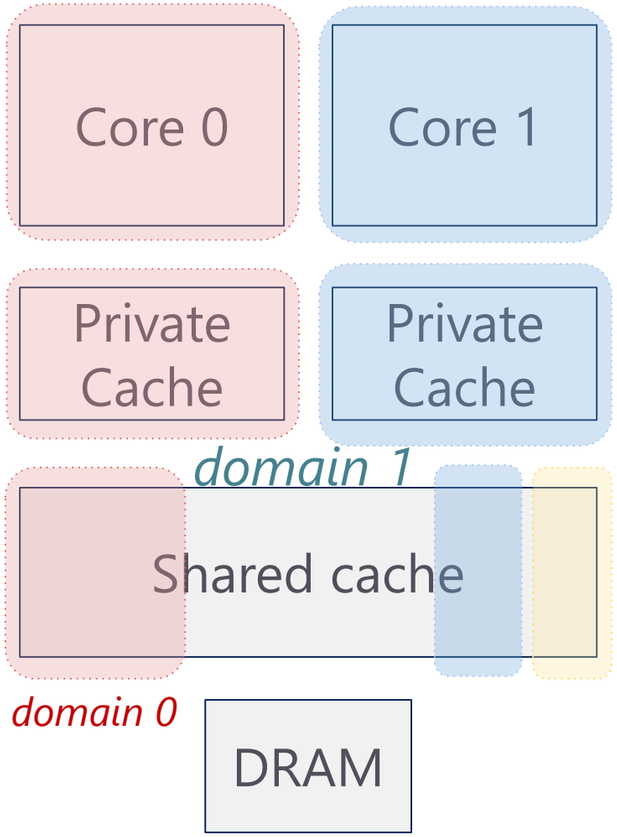
# Defense Schema




**Block any of these steps!**

1. Create a channel
2. Create the transmitter
3. Launch the transmitter
4. Access the secret

# Intel's Cache Allocation Technology (CAT)

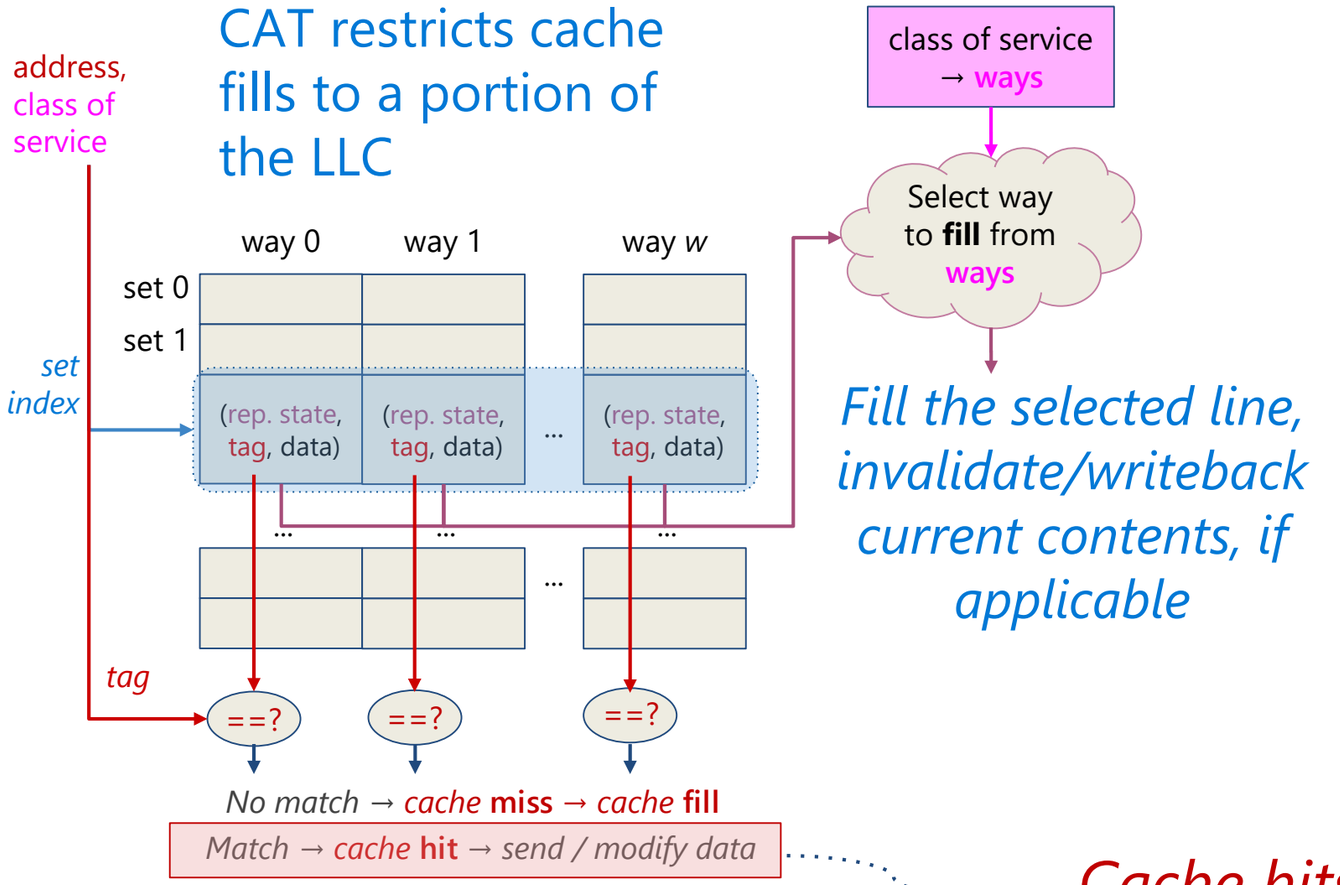



CAT can be configured to prevent a **potential transmitter** from evicting LLC lines of a potential receiver.

Way partitioning is flexible, but CAT is built for QoS and not for security

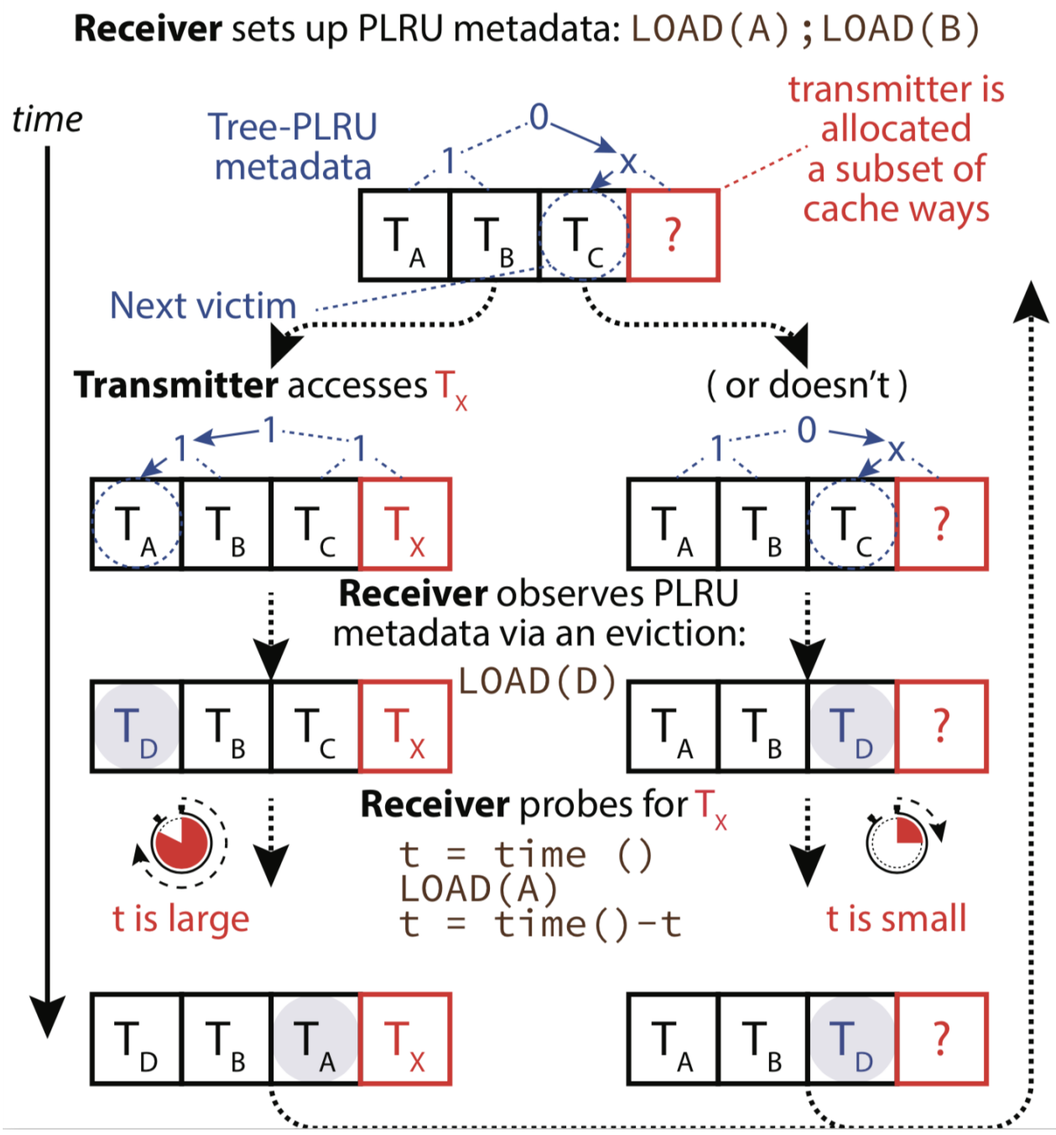
- Shared addresses are visible across domains
- Replacement metadata updates are not isolated

# Intel's CAT leaks information through cache hits



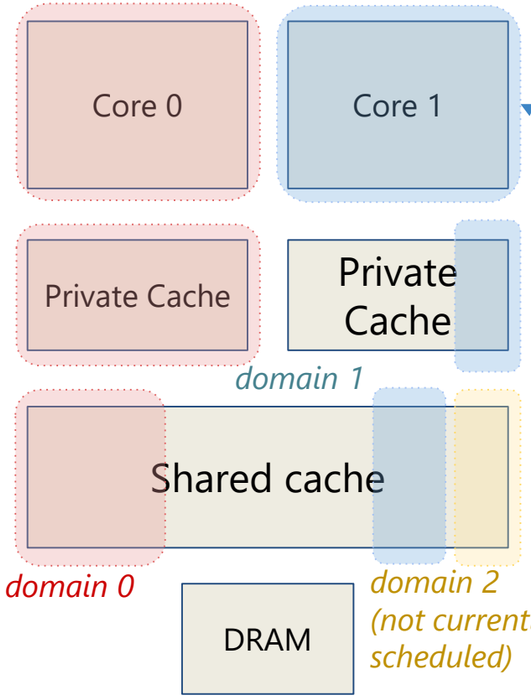

*Cache hits are not isolated across domains*

# Sharing replacement metadata leaks information



# *Our Work*: DAWG: Dynamically Allocated Way Guards

DAWG tracks *global* **protection domains**




Caches ensure protection domains do not interfere via cache tags or replacement metadata.

Cores tag each access with a protection domain id:

Core 1's DAWG domain_id MSR

| Instruction fetch domain | Load domain | Store* domain |
|---|---|---|

Need DAWG-like approach for other shared microarchitectural state, e.g., branch predictors

CSAIL

# Complication!

Masking cache hits may lead to **duplicated lines**!

→ OS ensures only clean, read-only lines are duplicated.

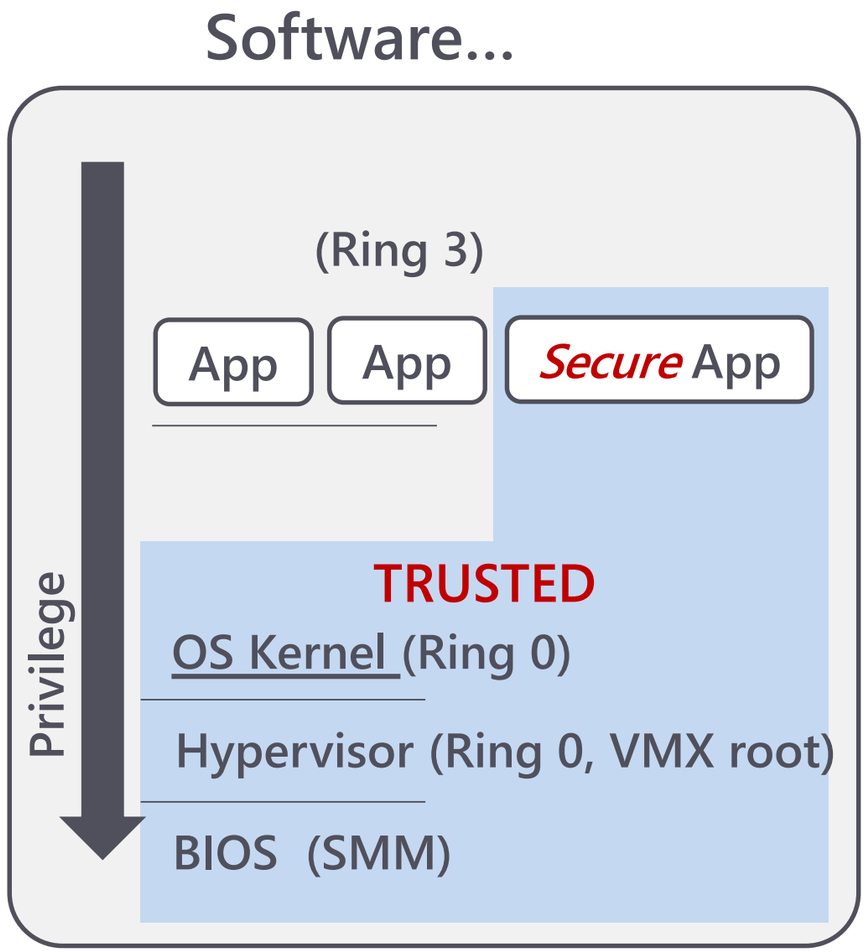
This is conveniently compatible with modern copy-on-write sharing

- Efficient ways to handle MMAP and Fork

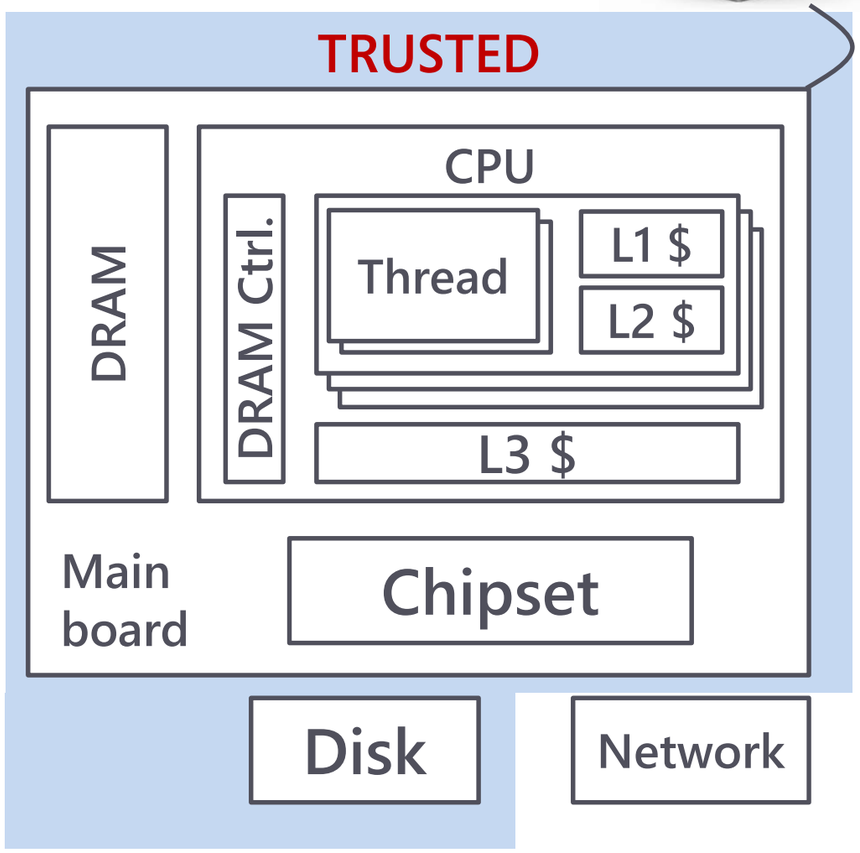
# Outline

- Violating isolation by exploiting speculative execution
- Defenses against cache timing attacks
- **Secure enclaves in Intel SGX and MIT Sanctum**

# A Typical Computer System' TCB



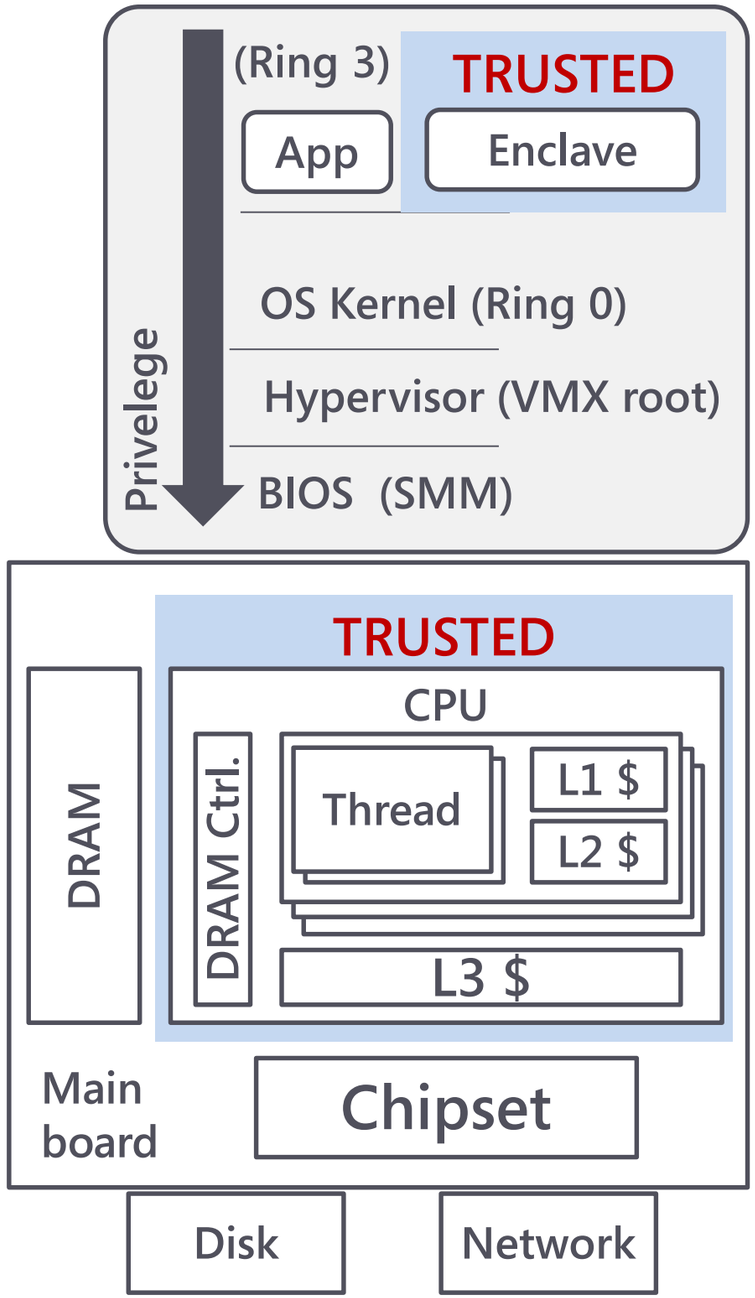

… Running on hardware




CSAIL

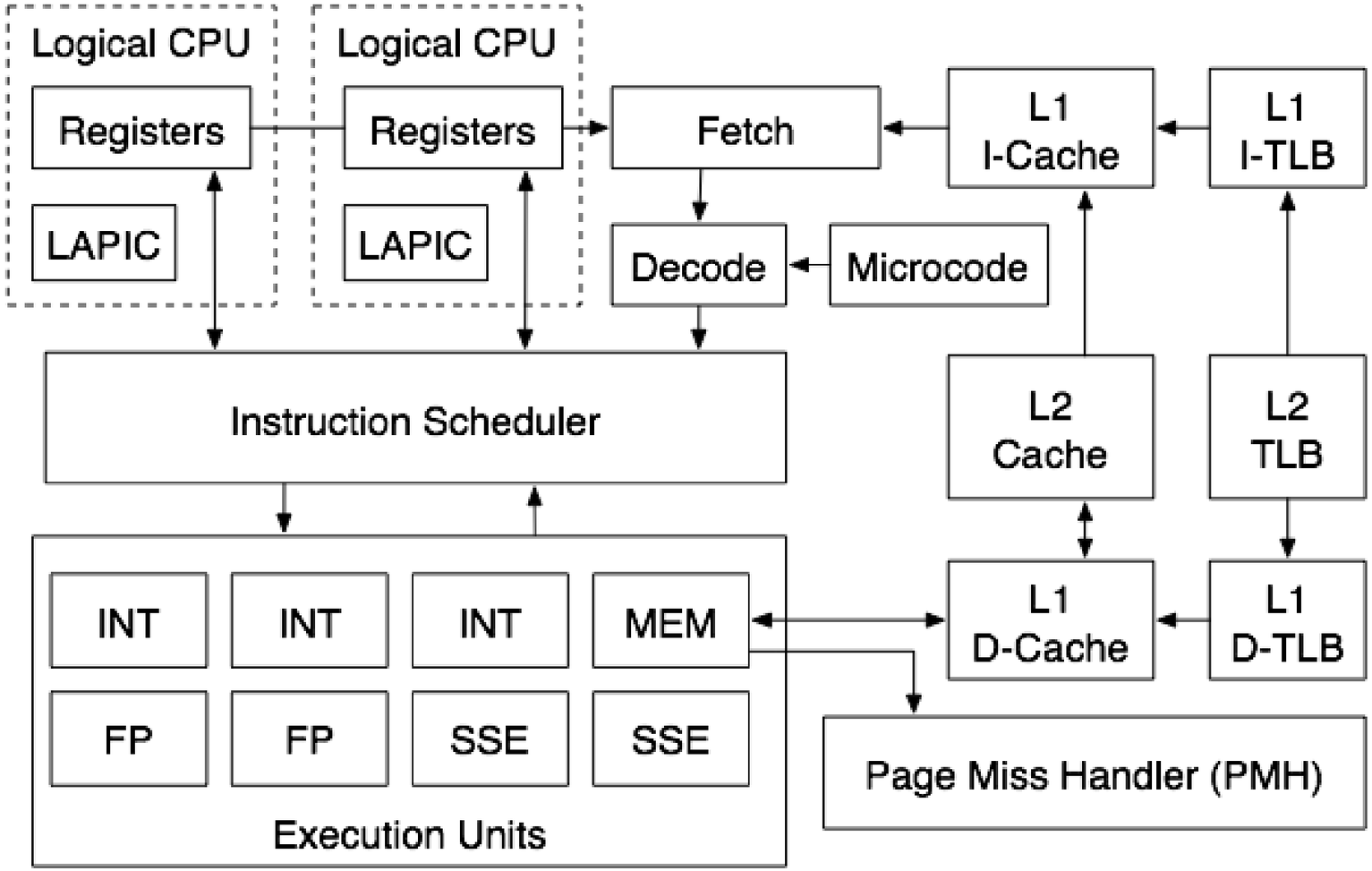
# Intel's SGX to reduce TCB

- SGX protects a **small** codebase
  - good!
- Protected process = "*Enclave*"
- Provides a trusted environment:
  - app integrity
  - protects data



# SGX leaks privacy in many ways

## Hyperthreading, Speculation, Page Tables, Caches, …



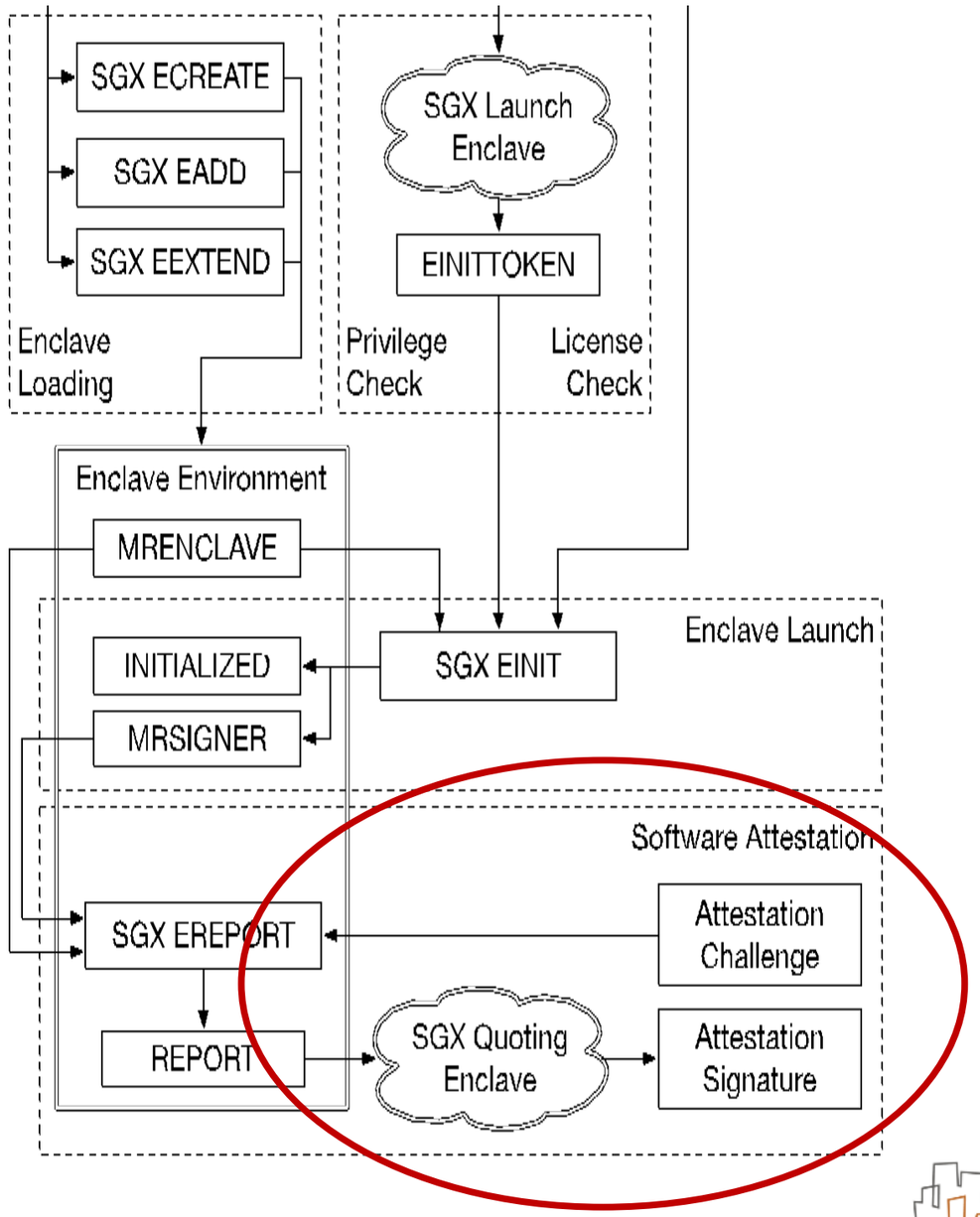

CSAIL

# SGX Uses Enclaves for Attestation (EPID)

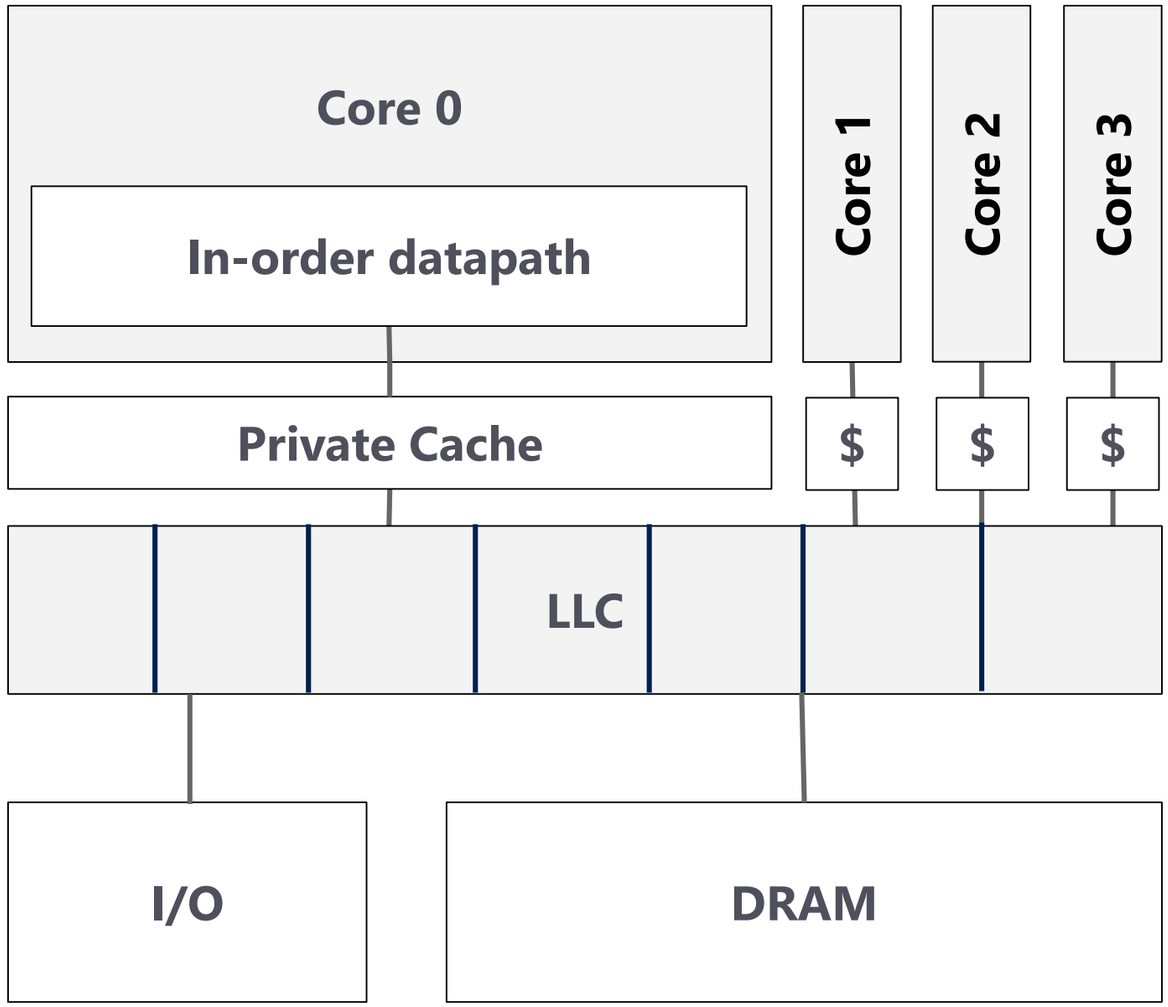
Software uses attestation key to sign results of computation

Cache timing attacks could leak the key

**Foreshadow, Usenix Security**

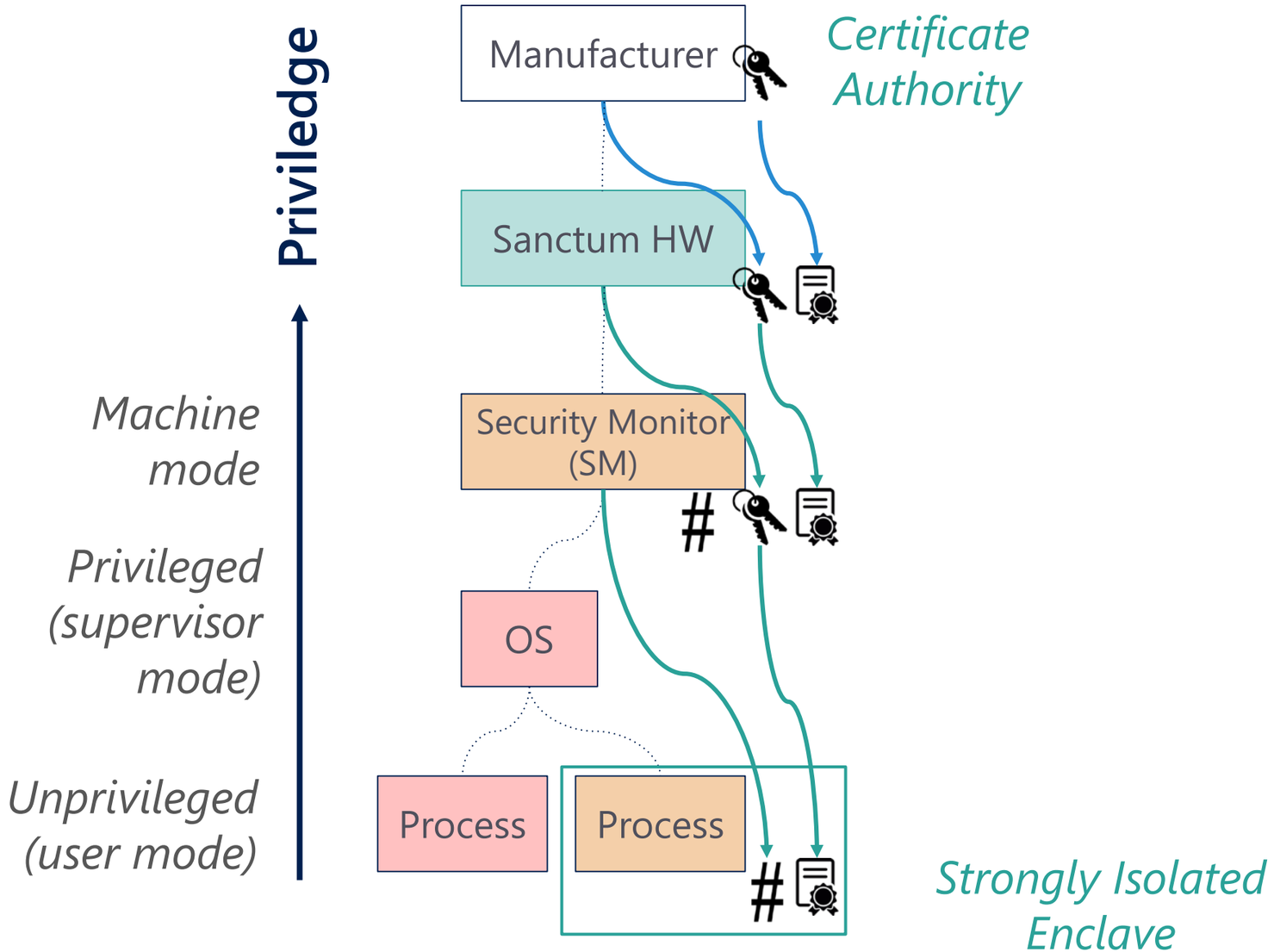


# Sanctum Secure Processor
## No Speculation, No Hyperthreading

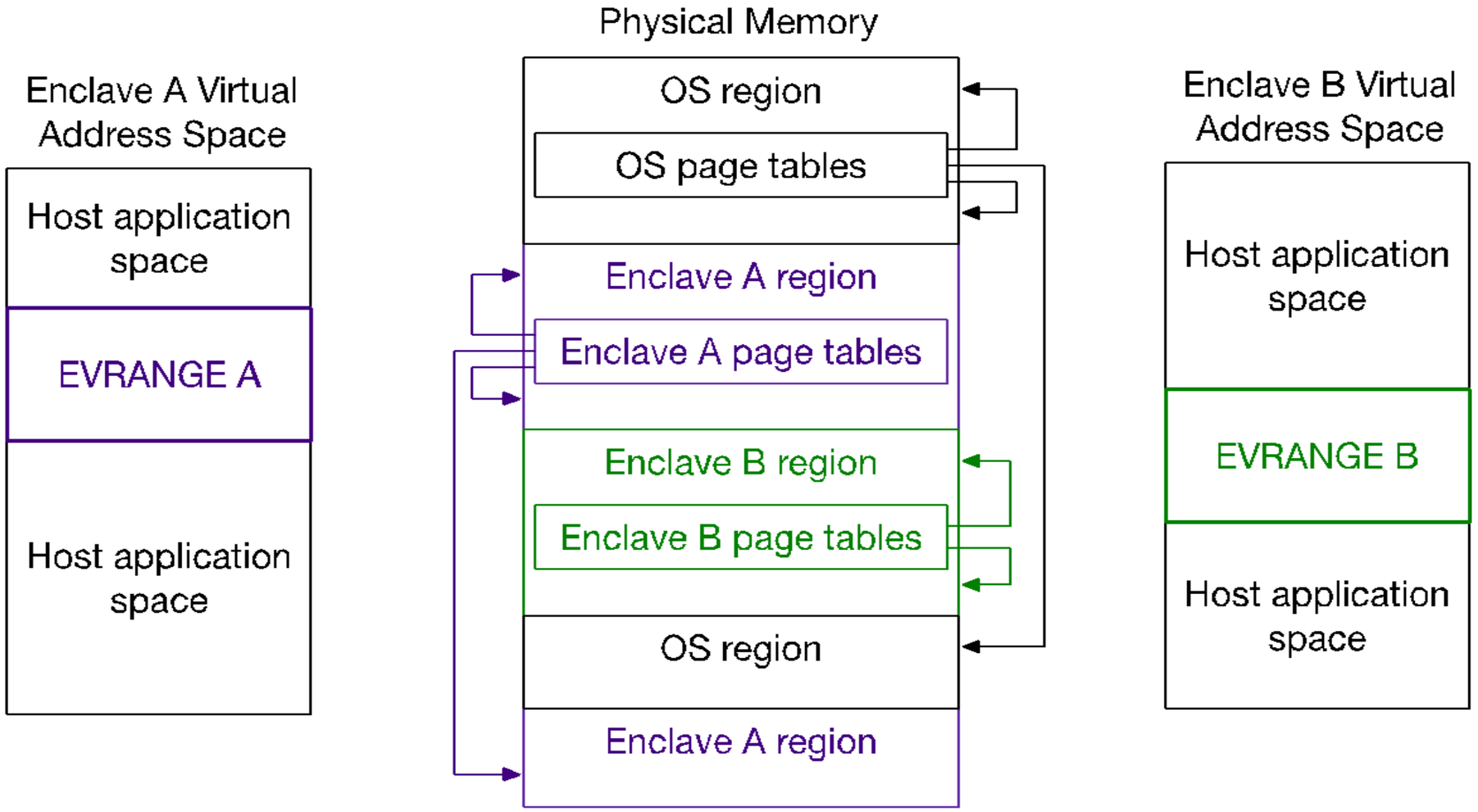



Partitioned Last Level Cache

# Sanctum's Chain of Trust

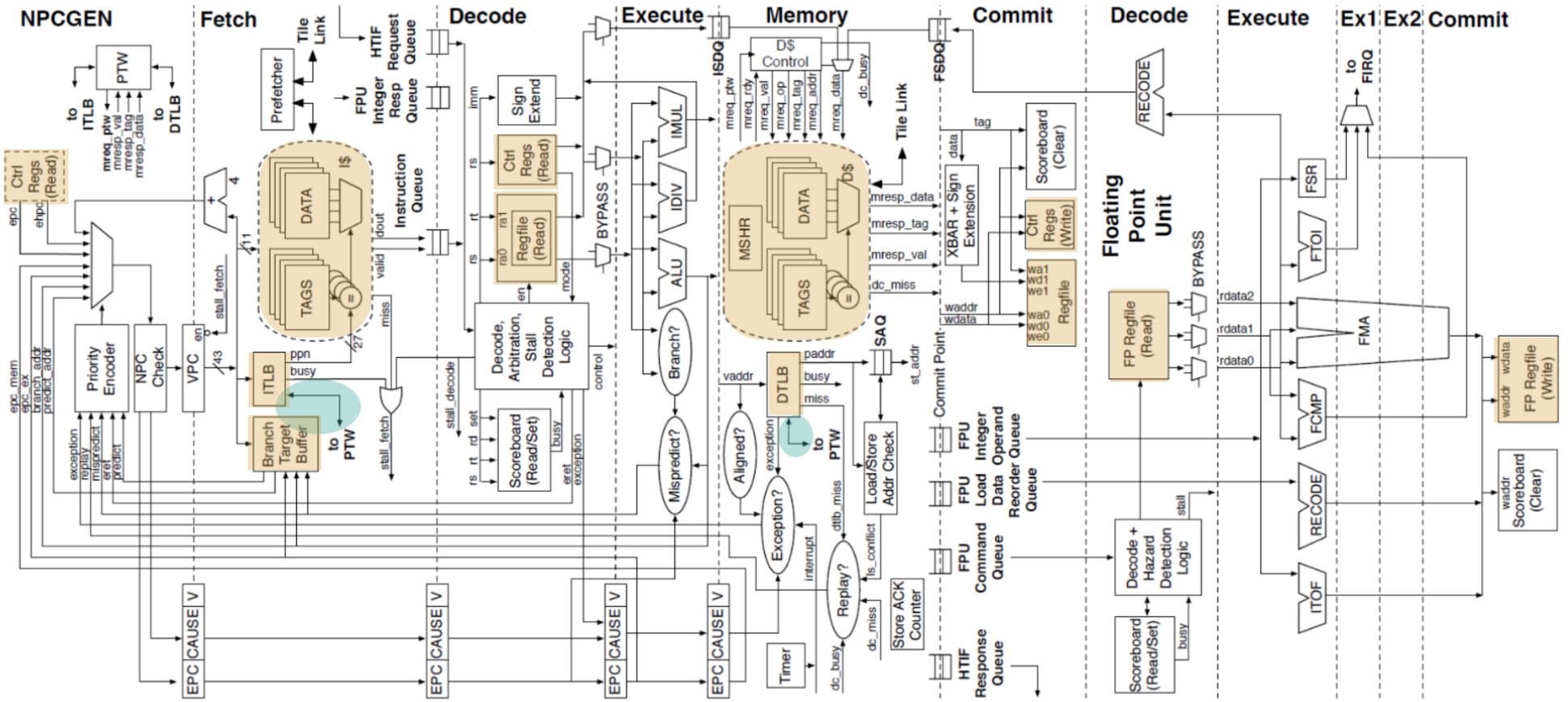


# Isolated Page Tables



# Sanctum Secure Processor
# No Speculation, No Hyperthreading

RISCV Rocket Core, Changes required by Sanctum (+ ~2% of core)




Also requires 9 new config registers

CSAIL

# Status

- Sanctum on AWS F1—you too can use it (or break it!)
- *Ongoing*: Keystone processor on HiFive Unleashed RISC-V chip (with Krste Asanovic and Dawn Song, UCB)
- *Near future*: Out-of-order "Sanctoom" processor
- *Near future*: Formal verification effort (with Adam Chlipala, MIT)

# In Conclusion,

- Significant security concerns with outsourcing computation especially to public clouds
- Intel’s SGX helps but leaks privacy through software side channels and is quite opaque
- Rethinking processor architecture to not sacrifice isolation and privacy when optimizing for performance

# Thank you!

Microsoft